IN THE CLAIMS

1. (currently amended) A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of processing units coupled to and for accessing a shared memory, the method comprising:

providing that selected processor tasks for execution and having respective priorities be copied from the shared memory to one or more of the processing units, and that each of the selected tasks is executed at one of the processing units;

determining a second processing unit from the plurality of the processing units except for a first of the processing units, wherein the second processing unit is executing a processor task of lowest priority and having a lower priority than a first selected processor task being executed at the first processing unit; and

migrating the first selected~~at least one~~ processor task being executed at the~~a~~ first ~~of the~~ processing unit~~s~~ from the first processing unit~~, based on priority of the selected tasks,~~ to the second ~~another of the~~ processing unit~~s~~.

2. (currently amended) The method of claim 1, further comprising prohibiting ~~the~~ execution of the first selected processor task from the shared memory after ~~the~~ copying of the first selected processor task to one or more of the processing units.

3. (original) The method of claim 1, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the processor tasks are copied to local memory and executed in local memory.

4. (currently amended) The method of claim 1, wherein migration of the first ~~at least one~~ selected processor task is based on a condition.

5. (canceled)

6. (currently amended) The method of claim 4~~5~~, wherein satisfaction of the condition and the initiating of the migration is not based on preemptive action.

7. (currently amended) The method of claim 3~~5~~, further comprising: requiring that the sub-processing units select processor tasks from the shared memory for execution based on their priority levels.

8. (original) The method of claim 3, further comprising: requiring that the sub-processing units select a processor task of higher priority before a processor task of lower priority from the shared memory.

9. (previously presented) The method of claim 3, further comprising:

selecting a first processor task of a first priority level from the shared memory for execution by a first sub-processing unit;

selecting a second processor task of a second priority level from the shared memory for execution by a second sub-processing unit; and

yielding the first sub-processing unit to a third processor task of a third priority level before completing the execution of the first processor task, the third processor task being selected because its priority level is higher than any other processor tasks that are ready to be executed.

10. (currently amended) A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of processing units coupled to and for accessing a shared memory, the method comprising:

providing that selected processor tasks for execution and having respective priorities be copied from the shared memory to one or more of the processing units, and that each of the selected tasks is executed at one of the processing units;

providing that the processing units select processor tasks from the shared memory for execution based on the priority levels of the processor tasks;

determining a second processing unit from the plurality of the processing units except for a first of the processing units, wherein the second processing unit is running a processor task of lowest priority and having a lower priority than a first selected processor task running on the first processing unit;

migrating the first ~~a~~ selected processor task ~~of lower priority~~ running on the ~~a~~ first ~~of the~~ processing unit~~s~~ from the first processing unit~~, based on priority of the selected tasks,~~ to the second ~~another of the~~ processing unit~~s~~; and

after the migrating, providing that the first processing unit run a processor task having a higher priority than the migrated~~,~~ first selected ~~lower prior~~ processor task.

11. (currently amended) The method of claim 10, further comprising prohibiting the execution of the first selected processor task from the shared memory after ~~the~~ copying of the processor task to one or more of the processing units.

12. (original) The method of claim 10, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the processor tasks are copied to local memory and executed in local memory.

13. (original) The method of claim 12, further comprising: requiring that the sub-processing units select a processor task of higher priority before a processor task of lower priority from the shared memory.

14. (previously presented) The method of claim 12, further comprising:

selecting a plurality of processor tasks of associated priority levels from the shared memory, based on the priority levels, for execution by a number of sub-processing units;

causing an n-th processor task in the shared memory having a given priority level to become ready for execution; and

determining whether the given priority level is higher than any of the priority levels of the plurality of processor tasks whose execution has not been completed.

15. (original) The method of claim 14, wherein at least one of the sub-processing units is operable to perform the determination.

16. (original) The method of claim 14, further comprising: preemptively replacing one of the plurality of processor tasks of lower priority level than the given priority level with the n-th processor task.

17. (previously presented) The method of claim 16, wherein a first of the sub-processing units is operable to at least initiate the replacement and cause another of the plurality of sub-processing units to yield execution of a processor task of lower priority level.

18. (previously presented) The method of claim 17, further comprising: providing that the initiating first sub-processing unit issues an interrupt to the yielding, another sub-processing unit in order to initiate the replacement of the processor task of lower priority level.

19. (original) The method of claim 17, further comprising: providing that the yielding sub-processing unit writes the processor task of lower priority from its local memory back into the shared memory.

20. (original) The method of claim 17, further comprising: providing that the yielding sub-processing unit copies the n-th processor task of higher priority from the shared memory into its local memory for execution.

21. (currently amended) A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of sub-processing units and a main processing unit coupled to and for accessing a shared memory, each sub-processing unit including an on-chip local memory separate from the shared memory, the method comprising:

providing that the processor tasks for execution and having respective priorities be copied from the shared memory into the local memory of the sub-processing units and that each of the tasks is executed at one of the sub-processing units, and prohibiting the execution of the processor tasks from the shared memory after the copying into the local memory of the sub-processing units;

selecting a plurality of processor tasks of associated priority levels from the shared memory for execution by one or more of the sub-processing units;

providing that the sub-processing units determine whether an n-th processor task in the shared memory having a given priority level has a higher priority level than any of the priority levels of the plurality of processor tasks; ~~and~~

determining a second sub-processing unit from the plurality of the sub-processing units except for a first of the sub-processing units, wherein the second sub-processing unit is executing a processor task of lowest priority and having a lower priority than a first selected processor task being executed on the first sub-processing unit; and

migrating the first ~~at least one~~ processor task being executed at the ~~a~~ first ~~of the~~ sub-processing unit~~s~~ from the first sub-processing unit ~~, based on priority of the task being executed at the first sub-processing unit,~~ to ~~another of~~ the second sub-processing unit~~s~~.

22. (previously presented) The method of claim 21, further comprising: providing that a processor task of lower priority

running on one of the sub-processing units is preemptively replaced with a processor task of higher priority.

23. (original) The method of claim 21, further comprising: providing that the sub-processing units use a shared task priority table in determining whether the n-th processor task is of a higher priority level than the plurality of processor tasks.

24. (previously presented) The method of claim 23, wherein: the shared task priority table includes entry pairs of sub-processing unit identifiers and processor task priority identifiers; and each entry includes a sub-processing unit identifier and priority identifier pair that indicate a priority level of a given processor task running on an associated sub-processing unit.

25. (previously presented) The method of claim 21, further comprising: providing that a sub-processing unit, when seeking to determine whether the n-th processor task is of a higher priority level than the plurality of processor tasks, searches the shared task priority table to find an entry pair indicating a lower priority level.

26. (currently amended) A method of executing processor tasks on a multi-processing system, the multi-processing system including a plurality of sub-processing units and a main processing unit coupled to and for accessing a shared memory, each processing unit including an on-chip local memory separate from the shared memory, the method comprising:

providing that the processor tasks for execution and having respective priorities be copied from the shared memory into the local memory of the sub-processing units and that each of the tasks is executed at one of the sub-processing units, and prohibiting the execution of the processor tasks from the shared memory after the copying into the local memory of the sub-processing units;

providing that the sub-processing units select processor tasks from the shared memory for execution based on priority levels of the processor tasks;

~~-~~determining a second sub-processing unit from the plurality of the sub-processing units except for a first of the sub-processing units, wherein the second sub-processing unit is executing a processor task of lowest priority and having a lower priority than a first selected processor task running on the first sub-processing unit; and

migrating ~~a~~ the first selected processor task running ~~of higher priority running~~ on ~~a given one of~~ the first sub-processing units to ~~another of~~ the second sub-processing unit~~s running a processor task of lower priority~~ in response to an interrupt received by the first ~~given~~ sub-processing unit.

27. (currently amended) A multi-processor apparatus, comprising:

a plurality of processing units, each processing unit including local memory in which to execute processor tasks; and

a shared memory operable to store processor tasks that are ready to be executed and have respective priorities, wherein:

the processor tasks are copied from the shared memory into the local memory of the processing units for execution of the processor tasks by the processing units, and

a first ~~t least one~~ selected processor task being executed at a first of the processing units is migrated from the first processing unit~~, based on priority of the tasks ready to be executed,~~ to a second ~~another~~ of the processing units, wherein the second processing unit is determined from the plurality of the processing units except for the first processing unit, and wherein the second processing unit is executing a processor task of lowest priority and hav~~s~~ing a lower priority than the first selected processor task.

28. (currently amended) The apparatus of claim 27, further comprising prohibiting the execution of the first processor selected task from the shared memory after the copying of the first selected processor task to the local memory of one or more of the processing units.

29. (currently amended) The ~~method~~ apparatus of claim 27, wherein the plurality of processing units comprises a main processor unit and a plurality of sub-processing units, each of the plurality of sub-processing units having a local memory, and wherein the processor tasks are copied~~s~~ to the local memory and executed in the local memory.

30. (currently amended) The apparatus of claim 29, wherein migration of the first ~~at least one~~ selected processor task is based on a condition.

31. (canceled)

32. (currently amended) The apparatus of claim 30~~1~~, wherein satisfaction of the condition and the initiating of the migration is not based on preemptive action.

33. (currently amended) The apparatus of claim 30~~1~~, wherein the sub-processing units are operable to select processor tasks from the shared memory for execution based on their priority levels.

34. (original) The apparatus of claim 29, wherein the sub-processing units are operable to select a processor task of higher priority before a processor task of lower priority from the shared memory.

35. (previously presented) The apparatus of claim 29, wherein:

a first sub-processing unit is operable to select a first processor task of a first priority level from the shared memory for execution;

a second sub-processing unit is operable to select a second processor task of a second priority level from the shared memory for execution; and

the first sub-processing unit is operable to yield to a third processor task of a third priority level before completing the execution of the first processor task, the third processor task being selected because its priority level is higher than any other processor tasks that a ready to be executed.

36. (previously presented) The apparatus of claim 29, wherein the sub-processing units are operable to:

select a plurality of processor tasks of associated priority levels from the shared memory for execution; and

determine whether an n-th processor task in the shared memory having a given priority level that has become ready for execution at one of the sub-processing units has a higher level priority than any of the priority levels of the plurality of processor tasks whose execution has not been completed.

37. (original) The apparatus of claim 36, wherein at least one of the sub-processing units is operable to perform the determination.

38. (original) The apparatus of claim 36, wherein at least one of the sub-processing units is operable to preemptively replace one of the plurality of processor tasks of lower priority level than the given priority level with the n-th processor task.

39. (previously presented) The apparatus of claim 38, wherein a first of the sub-processing units is operable to at least initiate the replacement and cause another of the plurality of sub-processing units to yield execution of a processor task of lower priority level.

40. (original) The apparatus of claim 39, wherein the initiating sub-processing unit is operable to issue an interrupt

to the yielding sub-processing unit in order to initiate the replacement of the processor task of lower priority level.

41. (original) The apparatus of claim 39, wherein the yielding sub-processing unit is operable to write the processor task of lower priority from its local memory back into the shared memory.

42. (original) The apparatus of claim 39, wherein the yielding sub-processing unit is operable to copy the n-th processor task of higher priority from the shared memory into its local memory for execution.

43. (currently amended) A multi-processor apparatus, comprising:

a plurality of sub-processing units, each sub-processing unit including an on-chip local memory and for executing processor tasks; and

a shared memory operable to store processor tasks having respective priorities and that are ready to be executed, wherein:

the processor tasks are copied from the shared memory into the local memory of the sub-processing units for execution by the sub-processing units, and the processor tasks are not executed from the shared memory,

the sub-processing units are operable to select processor tasks from the shared memory for execution based on the priority levels of the processor tasks; and

at least a first ~~one~~ of the sub-processing units is operable to migrate a first selected processor task ~~of higher priority~~ running on ~~a given one of~~ the first sub-processing units to a second ~~another~~ of the sub-processing units ~~running a processor task of lower priority~~ in response to an interrupt received by the first ~~given~~ sub-processing unit, wherein the second sub-processing unit is determined from the plurality of the sub-processing units except for the first sub-processing

unit, and wherein the second processing unit is running a second processor task of lowest priority and having a lower priority than the first selected processor task.

44. (original) The apparatus of claim 43, wherein the sub-processing units are operable to select a processor task of higher priority before a processor task of lower priority from the shared memory.

45. (currently amended) The apparatus of claim 43, wherein the sub-processing units are operable to: select a plurality of processor tasks of associated priority levels from the shared memory for execution; and determine which of the plurality of processor tasks running on the sub-processing units has a lowest priority level that is lower than the priority level of the first selected processor task running on the ~~given~~ first sub-processing unit.

46. (currently amended) The apparatus of claim 45, wherein the ~~given~~ first sub-processing unit is operable to perform the determination.

47. (currently amended) The apparatus of claim 45, wherein the ~~given~~ first selected processor task is migrated to the second sub-processing unit running the second processor task ~~of lowest priority level~~ and replaces ~~ing~~ the second ~~at~~ processor task at the second sub-processing unit.

48. (currently amended) The apparatus of claim 47, wherein the ~~given~~ first sub-processing unit is operable to at least initiate the migration and cause~~ing~~ the second sub-processing unit running the second processor task ~~of lowest priority level~~ to yield execution to the first selected ~~given~~ processor task ~~of higher priority level~~.

49. (currently amended) The apparatus of claim 48, wherein the ~~given~~ first sub-processing unit is operable to issue an interrupt to the second ~~yielding~~ sub-processing unit in order to

initiate the replacement of the second processor task ~~of lowest priority level~~.

50. (currently amended) The apparatus of claim 48, wherein the ~~yielding~~ second sub-processing unit is operable to write the second processor task ~~of lowest priority~~ from its local memory back into the shared memory.

51. (currently amended) The apparatus of claim 45, wherein the ~~yielding~~ second sub-processing unit is operable to copy the first selected ~~given~~ processor task ~~of higher priority~~ from the local memory of the first ~~given~~ sub-processing unit into its local memory for execution.

52. (currently amended) The apparatus of claim 43, wherein the ~~given~~ first sub-processing unit is operable to use a shared task priority table in determining which processor task is of the lowest priority level.

53. (previously presented) The apparatus of claim 52, wherein: the shared task priority table includes entry pairs of sub-processing unit identifiers and processor task priority identifiers; and each entry includes a sub-processing unit identifier and priority identifier pair that indicate a priority level of a given processor task running on an associated sub-processing unit.

54. (currently amended) The apparatus of claim 53, wherein the ~~given~~ first sub-processing unit is operable to search the shared task priority table to find an entry pair indicating a lowest priority level.

55. (original) The apparatus of claim 53, wherein the sub-processing units are operable to modify the shared task priority table such that the entry pairs are current.